**Amendments to the Specification:**

Please replace the paragraph beginning on page 2, line 9 with the following:

Each master of a Bluetooth piconet can switch contexts from one slave to another. In addition, because a slave can participate in more than one piconet, a slave can switch ~~contests~~ contexts among masters. Traditional approaches render the context-switch process slow and cumbersome. What is needed is a system and method for performing relatively rapid context switches while still retaining an acceptable level of network throughput.

Please replace the paragraph beginning on page 3, line 18 with the following:

At least one embodiment of the peripheral system further includes a state machine, where the state machine includes an address portion, a control portion, and a data portion, where the first and second registers are included in the data portion. At least one embodiment of the address portion includes a register access circuit. At least one other embodiment of the peripheral system further includes a microprocessor. At least one other embodiment of the peripheral system includes at least one index register. At least one other embodiment of the peripheral system includes a plurality of context registers,

with each of the plurality of context registers being associated with one of a plurality of index values.

Please replace the paragraph beginning on page 4, line 11 with the following:

**FIG. 7** is a data flow diagram illustrating at [[lest]] least one embodiment of the data flow during the use of a master index to perform context-switching for a slave device.

Please replace the paragraph beginning on page 4, line 14 with the following:

**FIG. 8** illustrates a data flow diagram illustrating at [[lest]] least one embodiment of the data flow during the use of a slave index to perform context-switching for a master device.

Please replace the paragraph beginning on page 5, line 9 with the following:

**Figure 2** illustrates a network **20** that includes a plurality of wireless devices **102-1**, **102-2**. . . **102-i**. . . **102-n** ($2 \leq i \leq n$). Wireless network **20** is, for example, a point-to-multipoint Bluetooth piconet where wireless device **102-1** is a master Bluetooth system and wireless devices **102-2** through **102-n** are slave Bluetooth systems. As with all piconets, the master **102-1** and all the slaves **102-2** through **102-n** communicate over the same channel. In at least one embodiment, up to seven slaves can be active in the piconet **102**. One skilled in the art will recognize

that the number of active slaves supported in a piconet depends on many variables and design considerations. Any of a number of slaves can be supported. One skilled in the art will further recognize that a point-to-point network need not include Bluetooth devices **102-1**, **102-2**, but, rather, may comprise any type of computing device.

Please replace the paragraph beginning on page 5, line 21 with the following:

In addition to the active slaves **102-2**, **102-i** through **102-n** illustrated in **Figure 2**, a point-to-multipoint piconet **20** may include many additional slaves that can remain locked to the master **102-1** in a so-called "parked" state. When a slave does not need to participate on the piconet channel, but still needs to remain synchronized to the channel it can enter the low-power parked state. These parked slaves cannot be active on the piconet channel, but remain synchronized to the master. In at least one embodiment, up to 256 parked slaves may remain locked to the master. For additional information regarding an implementation of the synchronization of master and slave Bluetooth systems, please refer to U.S. Patent No. 6,650,880, ~~the commonly assigned application bearing U.S. Patent Application Serial No. ____________, entitled "Wireless Data Communications Using FIFO For Synchronization Memory," by inventors Sherman Lee, Vivian Y. Chou, and John H. Lin, filed June 12, 2000, attorney docket number M-8741 US,~~ which is herein incorporated by reference for all purposes.

Please replace the paragraph beginning on page 7, line 5 with the following:

The discussion above illustrates the need, in a network environment such as, for instance, a wireless network as specified in the Bluetooth Specification, for both masters and slaves to have the ability to quickly switch contexts in order to facilitate orderly, reasonably rapid communication among wireless devices. In a different context, several schemes exist for switching contents among software application programs. These schemes usually involve storing context information in memory and then, during a context switch, loading the information from memory into a particular register or set of registers. Such software-based context-switching schemes are disclosed, for example, in U.S. Pat. No. ~~6.061,711~~ 6,061,711, entitled "Efficient Context Saving and Restoring in a Multi-Tasking Computing System Environment", and issued to Song et al. [[an]] and also in U.S. Pat. No. 5,812,823, entitled "Method and System for Performing an Emulation Context Save and Restore That is Transparent to the Operating System", and issued to Kahle et al. Song '711 and Kahle '823 are hereby incorporated by reference for all purposes.

Please replace the paragraph beginning on page 7, line 19 with the following:

In a software-based context-switching scheme, a set of hardware registers is used to contain current context information. When a context switch occurs, the information in

the registers for outgoing context is stored from the registers to a memory location, such as a location in main memory or in a cache. The context information for the incoming context is then loaded from a different memory location into the registers. This storing to, and loading from, memory to transfer, via software, data between memory and hardware registers is time-consuming and processor-intensive. To employ such an approach for switching contexts among masters and slaves in a network of computing devices, such as a network of Bluetooth systems, would result in severely limited network throughput during the context-switch process.

Please replace the paragraph beginning on page 17, line 9 with the following:

**Figure 9** illustrates the organization of at least one embodiment of a Bluetooth system **900**, which is relevant to the following discussion. **Figure 9** illustrates that a Bluetooth system **900**, such as a cell phone or a PDA, consists of a host computer **970** and a peripheral system **972**. The peripheral system **972** includes an analog component and a digital component. The analog component is a radio unit **910**. For additional details concerning at least one aspect of at least one embodiment of the radio unit **910**, please refer to U.S. Patent No. 6,560,449, ~~co-pending application U.S. Pat. App. No. ______________, Attorney Docket No. M-8890 US, by inventor Albert Liu, entitled "Image Rejection I/Q Demodulators", filed June 12, 200,~~ which is herein incorporated by reference in its entirety for all purposes. For additional details concerning the operation of at least one

aspect of at least one other embodiment of the radio unit **910**, please refer to U.S. Patent No. 6,778,594, ~~co-pending application U.S. Pat. App. No. _______________, Attorney Docket Number M-8891 US, by inventor Albert Liu, entitled "Receiver Architecture Employing Low Intermediate Frequency And Complex Filtering", filed June 12, 2000,~~ which is herein incorporated by reference in its entirety for all purposes.

Please replace the paragraph beginning on page 19, line 24 with the following:

The discussion above illustrates that a master system having a slave index **42** and one or more slave registers **46** can quickly perform context-switching among slaves. Similarly, **Figure 7** illustrates that a slave system having a master index **40** and one or more master registers **44** can quickly perform context-switching among masters. Because the information described above in **Table 1** is stored in the slave registers **46**, and the information described above in **Table 2** is stored in the master registers **44**, the context information need not be stored in software data structures nor stored on the microprocessor of the host computer **970**. This allows the context data to be accessed more quickly than if it were transferred from memory to registers upon a context switch.

Please replace the paragraph beginning on page 21, line 9 with the following:

The Corereg register access circuit **70** receives various input values that are provided by software running on the host system **970**. One such input value is an Addr_in value, which represents the particular address of the desired field within the master register **44** or slave register **46** of interest. Another input value is the index value in either the master index register **40** or the slave index register **42**. For purposes of illustration, reference is made to **Figures 3** and **7** and **Tables 1** and **2**. For instance, a slave system **32** may switch contexts from a first master **102-1** (whose context information is tracked, say, in master register **44a**) to a second master **36** (whose context information is tracked, say, in master register **44b**). In at least one embodiment, an index value of 4b'0000' represents the first master register **44a** while an index value of 4b'0001' represents the second master register **44b**. In order to switch the contexts, then, [[then]] <u>the</u> link controller **920 (Figure 9)** replaces the index value of 4b'000' with an index value of4b'0001' in the master index register **40**. In this manner, a context switch has occurred.

Please replace the paragraph beginning on page 23, line 16 with the following:

**Figure 12** illustrates that the host computer **970**, when executing operation **1202**, loads the appropriate index value into the [[salve]] <u>slave</u> index register **42**. For instance, if the slave register of interest **46** is a second slave register (where the index value for a first slave register is zero), then a value of 4b'01' is loaded by the host **970** into the slave index

register **42** in operation **1202**. The host computer **970** then executes operation **1204**. Operation **1204** invokes the Corereg logic circuit **70 (Figure 7)** to write or read from the register of interest **46**. Operation **1204** includes a read path and a write path.

Please delete the paragraph beginning on page 25, line 3.